**How to Write an SRS Document**

Writing an SRS document is important. But it isn't always easy to do. Here are five steps you can follow to write an effective SRS document.

**1. Create an Outline (Or Use an SRS Template)**

Your first step is to create an outline for your software requirements specification. This may be something you create yourself. Or you may use an existing SRS template.

If you're creating this yourself, here's what your outline might look like:

1. Introduction

1.1 Purpose

1.2 Intended Audience

1.3 Intended Use

1.4 Scope

1.5 Definitions and Acronyms

2. Overall Description

2.1 User Needs

2.2 Assumptions and Dependencies

3. System Features and Requirements

    3.1 Functional Requirements

    3.2 External Interface Requirements

    3.3 System Features

    3.4 Nonfunctional Requirements

Once you have your basic outline, you're ready to start filling it out.

## 2. Start With a Purpose

The introduction to your SRS is very important. It sets the expectation for the product you're building.

So, start by defining the purpose of your product.

Intended Audience and Intended Use

Define who in your organization will have access to the SRS — and how they should use it. This may include developers, testers, and project managers. It could also include stakeholders in other departments, including leadership teams, sales, and marketing.

Product Scope

Describe the software being specified. And include benefits, objectives, and goals. This should relate to overall business goals, especially if teams outside of development will have access to the SRS.

Definitions and Acronyms

It's smart to include a risk definition. Avoiding risk is top-of-mind for many developers — especially those working on safety-critical development teams.

Here's an example. If you're creating a medical device, the risk might be the device fails and causes a fatality.

By defining that risk up front, it's easier to determine the specific requirements you'll need to mitigate it.

## 3. Give an Overview of What You'll Build

Your next step is to give a description of what you're going to build. Is it an update to an existing product? Is it a new product? Is it an add-on to a product you've already created?

These are important to describe upfront, so everyone knows what you're building.

You should also describe why you're building it and who it's for.

User Needs

User needs — or user classes and characteristics — are critical. You'll need to define who is going to use the product and how.

You'll have primary and secondary users who will use the product on a regular basis. You may also need to define the needs of a separate buyer of the product (who may not be a

primary/secondary user). And, for example, if you're building a medical device, you'll need to describe the patient's needs.

Assumptions and Dependencies

There might be factors that impact your ability to fulfill the requirements outlined in your SRS. What are those factors?

Are there any assumptions you're making with the SRS that could turn out to be false? You should include those here, as well.

Finally, you should note if your project is dependent on any external factors. This might include software components you're reusing from another project.

**4. Detail Your Specific Requirements**

The next section is key for your development team. This is where you detail the specific requirements for building your product.

Functional Requirements

Functional requirements are essential to building your product.

If you're developing a medical device, these requirements may include infusion and battery. And within these functional requirements, you may have a subset of risks and requirements.

External Interface Requirements

External interface requirements are types of functional requirements. They're important for embedded systems. And they outline how your product will interface with other components.

There are several types of interfaces you may have requirements for, including:

- User
- Hardware
- Software
- Communications

System Features

System features are types of functional requirements. These are features that are required in order for a system to function.

Other Nonfunctional Requirements

Nonfunctional requirements can be just as important as functional ones.

These include:

- Performance
- Safety
- Security
- Quality

The importance of this type of requirement may vary depending on your industry. Safety requirements, for example, will be critical in the medical device industry.

IEEE also provides guidance for writing software requirements specifications, if you're a member.

**5. Get Approval for the SRS**

Once you've completed the SRS, you'll need to get it approved by key stakeholders. And everyone should be reviewing the latest version of the document.